SOFTWARE ENGINEERING

DELIVERABLE-4

Group_4

Group Members:

1. Sadvik Kondadi
2. Sai Chandra Teja Akkala
3. Shashank Kodishala
4. Deepak Reddy Punuru
5. Deepu Gondhi
6. Padmaja Soma
7. Nithin Reddy Pinikesi
8. Sanjay Ramaswamy Adla
9. Thanwish Ram Pothugunta

1. Requirements

9. Recommended Courses for Specialization

Student Role

- Students see a **personalized course recommendation list** based on:
  - Courses they've already enrolled in.
  - Popular courses within their specialization or domain.

10. Professor Recommendation System

Student Role

- Students rate professors through spot evaluations (conducted after each lecture or assignment).
- Ratings include:
  - Very Interactive
  - Very Strict
  - Can score high
  - Very lenient in Correction
  - adds Grace Marks
  - Can learn a lot

- Other students have access to the aggregated survey insights (without seeing individual identities).

11. Payment & Subscription Management

Student Role

- Can **purchase courses**

12. Content Management

Student Role

- Access course **videos**, **PDFs**, **quizzes**, and **discussion boards**.
- Download course materials for offline study.

Instructor Role

- Upload and manage:
  - Videos (via YouTube links or file upload)
  - Reading materials (PDFs, PPTs)
  - Interactive quizzes (MCQs, Descriptive)
- Organize content into **modules**.

13. Assessment Enhancements

Student Role

- Attempt quizzes with:
  - **Shuffled MCQs** (reduces cheating).
  - **Descriptive questions** are evaluated by AI using similarity checking.

Instructor Role

- Upload questions with answer keys.
- View AI-evaluated results for descriptive questions.
- Can override AI grading if needed.

14. Attendance Tracking

Instructor Role

- Track **attendance percentage** per session or course.
- Generate **attendance reports** for grading or certification.

15. Feedback & Survey System

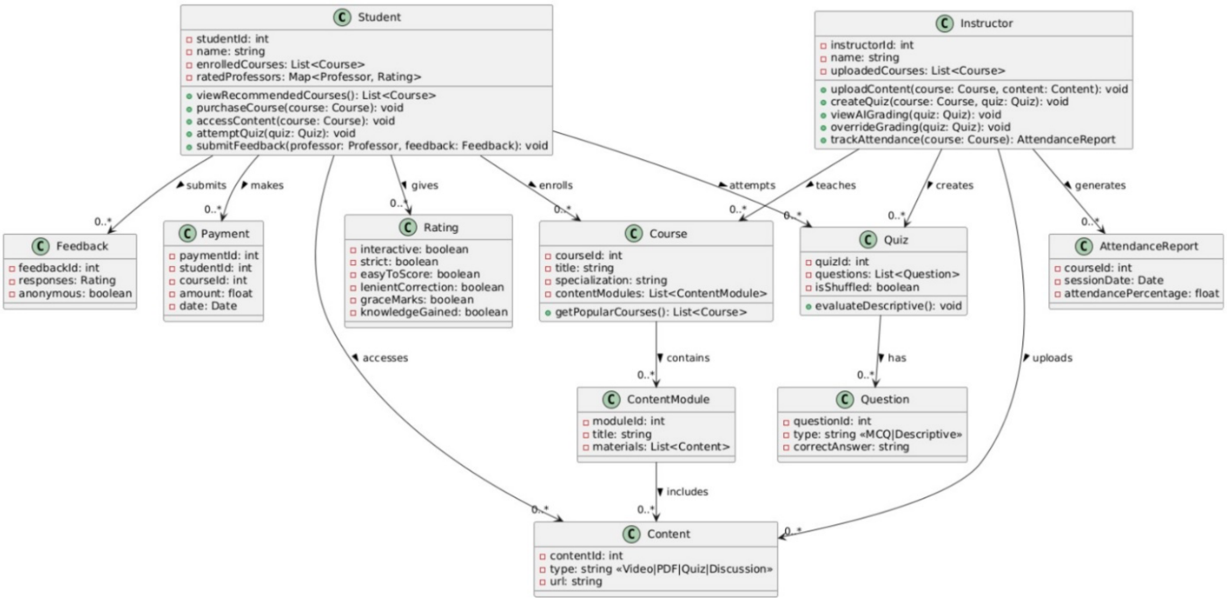
Student Role

- Fill out anonymous surveys about:

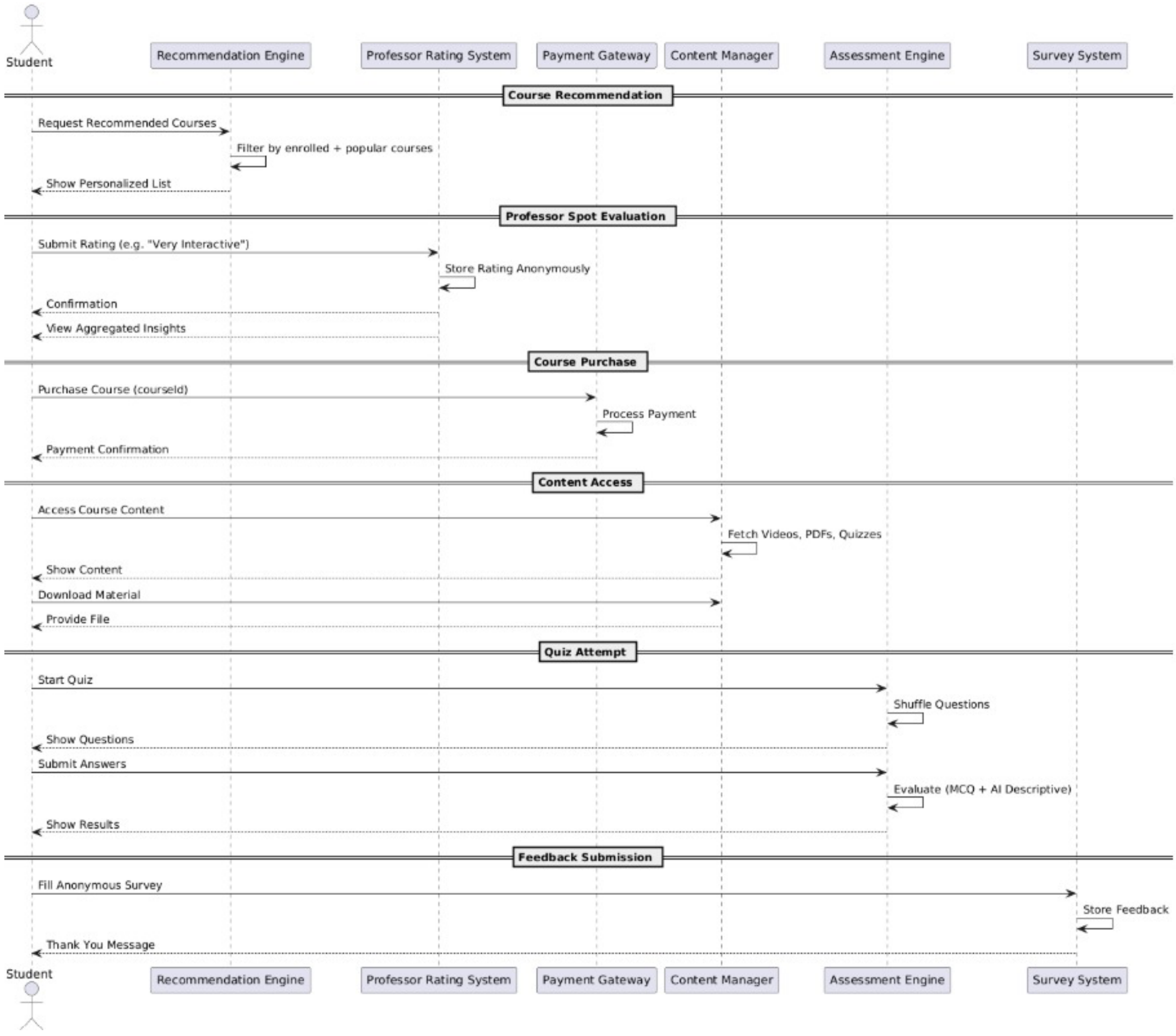
- Very Interactive
- Very Strict
- Can score high
- Very lenient in Correction
- adds Grace Marks
- Can learn a lot

2. UML Design

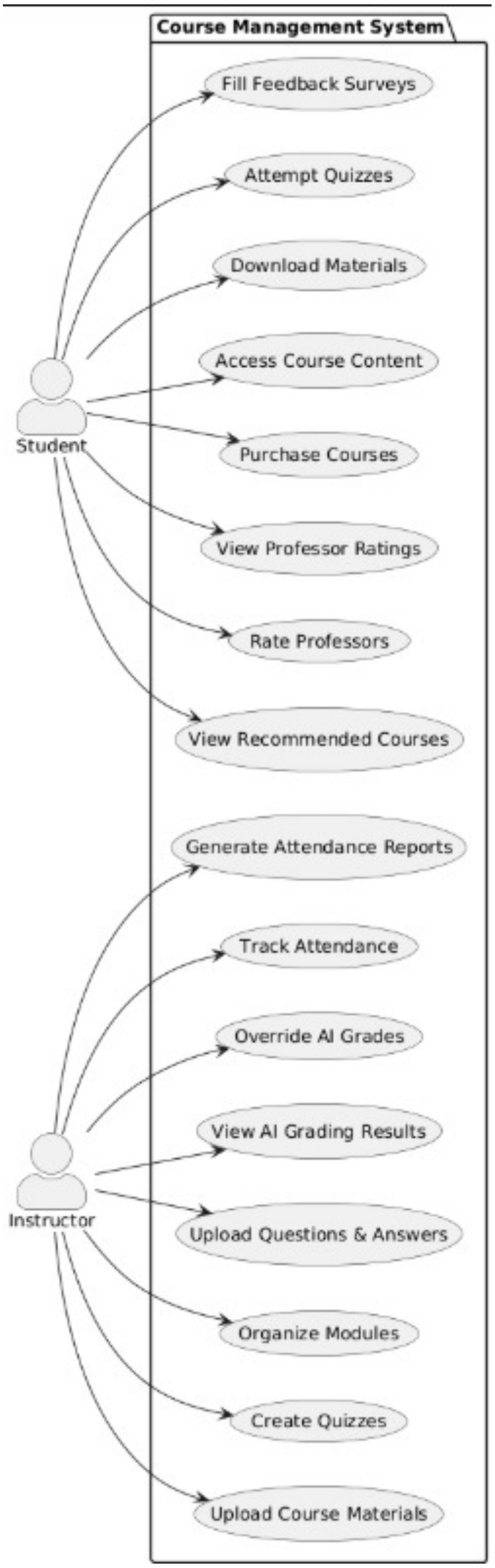
2.1 Class diagram


Student
studentId: int
name: string
enrolledCourses: List<Course>
ratedProfessors: Map<Professor, Rating>
viewRecommendedCourses(): List<Course>
purchaseCourse(course: Course): void
accessContent(course: Course): void
attemptQuiz(quiz: Quiz): void
submitFeedback(professor: Professor, feedback: Feedback): void
Instructor
instructorId: int
name: string
uploadedCourses: List<Course>
uploadContent(course: Course, content: Content): void
createQuiz(course: Course, quiz: Quiz): void
viewAIGrading(quiz: Quiz): void
overrideGrading(quiz: Quiz): void
trackAttendance(course: Course): AttendanceReport
submits
makes
gives
enrolls
attempts
teaches
creates
generates
0..*
Feedback
feedbackId: int
responses: Rating
anonymous: boolean
Payment
paymentId: int
studentId: int
courseId: int
amount: float
date: Date
Rating
interactive: boolean
strict: boolean
easyToScore: boolean
lenientCorrection: boolean
graceMarks: boolean
knowledgeGained: boolean
Course
courseId: int
title: string
specialization: string
contentModules: List<ContentModule>
getPopularCourses(): List<Course>
Quiz
quizId: int
questions: List<Question>
isShuffled: boolean
evaluateDescriptive(): void
AttendanceReport
courseId: int
sessionDate: Date
attendancePercentage: float
accesses
contains
has
uploads
ContentModule
moduleId: int
title: string
materials: List<Content>
Question
questionId: int
type: string «MCQ|Descriptive»
correctAnswer: string
includes
Content
contentId: int
type: string «Video|PDF|Quiz|Discussion»
url: string

2.2 Sequence diagram



2.3 Use case diagram



3. Test Cases Covered

test_slms_feedback_page_routing – Ensures navigation to the Feedback page triggers the feedback_ui component.

test_slms_course_recommendation_routing – Tests whether selecting Course Recom in the sidebar loads course_recommendation.

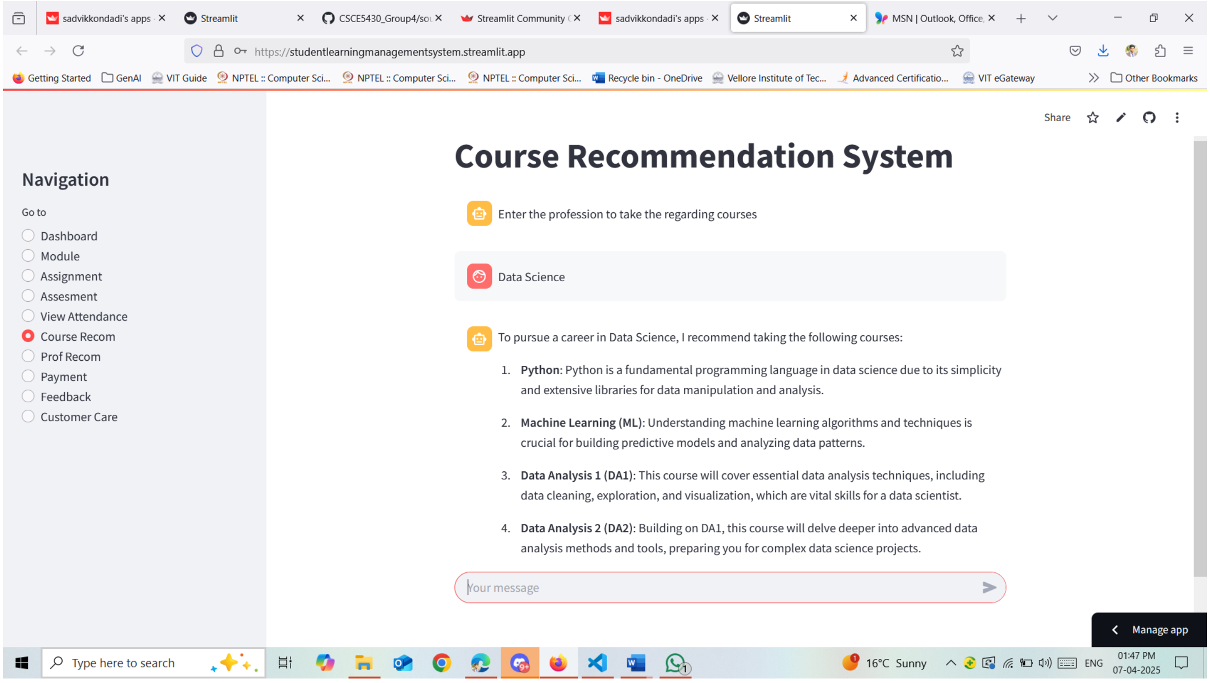
test_slms_professor_recommendation_routing – Validates that choosing Prof Recom triggers the prof_recommendation UI.

test_slms_view_attendance_routing – Checks if View Attendance in the sidebar routes to view_attendance_ui.

test_slms_roll_call_routing – Tests that selecting Roll Call launches the roll_call_ui function.

test_slms_payment_page_routing – Ensures the Payment option in the sidebar loads the payment_ui module.

4. User Manual



Navigation

Go to

○ Dashboard
○ Module
○ Assignment
○ Assesment
○ View Attendance
● Course Recom
○ Prof Recom
○ Payment
○ Feedback
○ Customer Care

Course Recommendation System

Enter the profession to take the regarding courses

Data Science

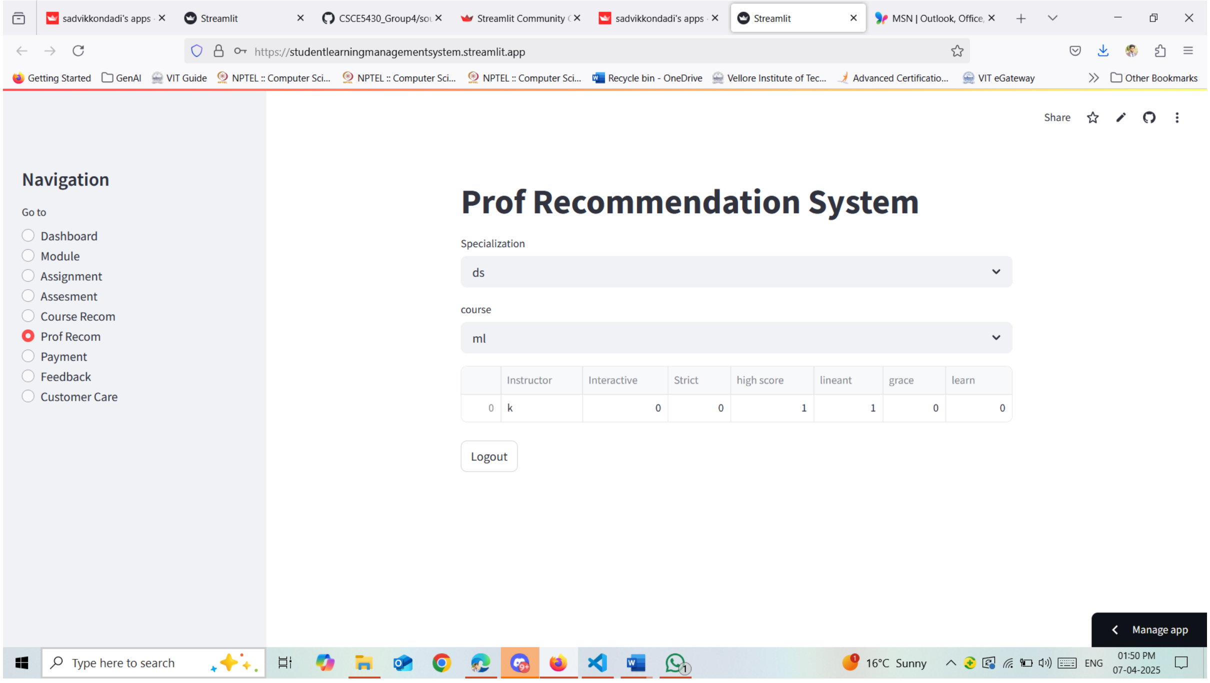
To pursue a career in Data Science, I recommend taking the following courses:

1. **Python**: Python is a fundamental programming language in data science due to its simplicity and extensive libraries for data manipulation and analysis.
2. **Machine Learning (ML)**: Understanding machine learning algorithms and techniques is crucial for building predictive models and analyzing data patterns.
3. **Data Analysis 1 (DA1)**: This course will cover essential data analysis techniques, including data cleaning, exploration, and visualization, which are vital skills for a data scientist.
4. **Data Analysis 2 (DA2)**: Building on DA1, this course will delve deeper into advanced data analysis methods and tools, preparing you for complex data science projects.

Your message

Course Recommendation

The default browser will open to the login page, which includes fields for user ID and password input, a button that redirects to the registration form, and a button to log in. If the entered credentials match the document present in the MongoDB database, then the user will be logged in, and the MongoDB documents have a role field along with the credentials, which the user gives. The domain of the role field is ‘Admin’, ‘Student’, and ‘Instructor’.



Navigation

Go to

○ Dashboard
○ Module
○ Assignment
○ Assesment
○ Course Recom
● Prof Recom
○ Payment
○ Feedback
○ Customer Care

Prof Recommendation System

Specialization: ds

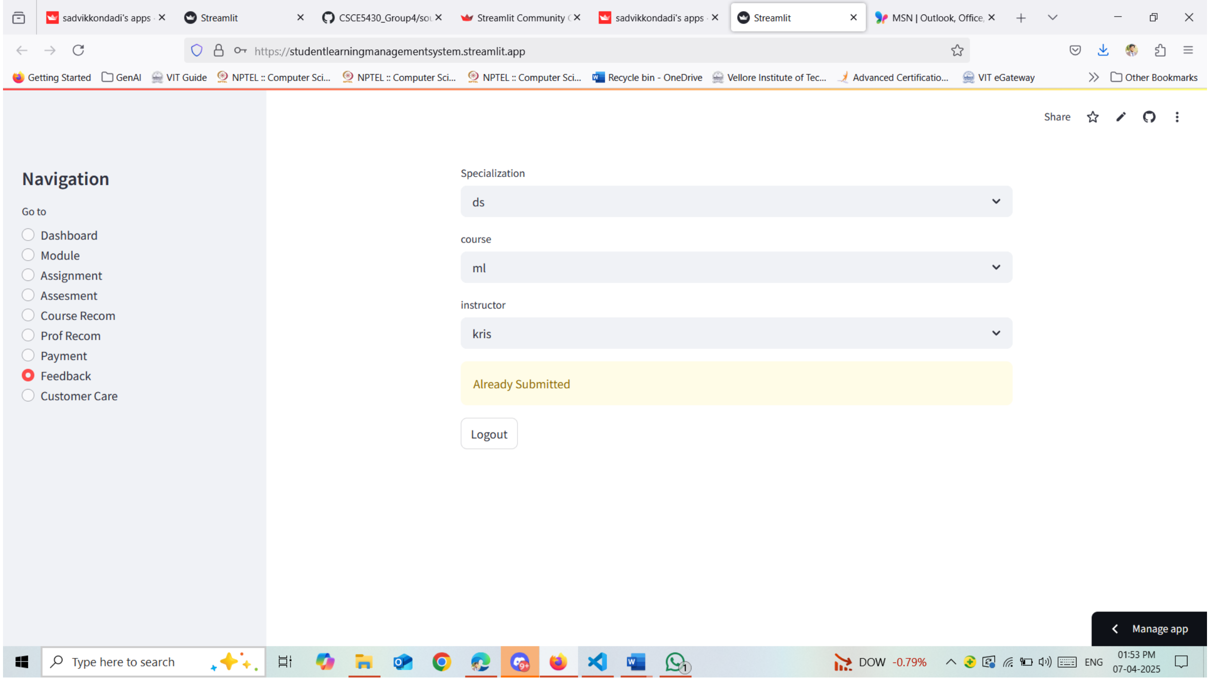
course: ml

	Instructor	Interactive	Strict	high score	lineant	grace	learn
0	k	0	0	1	1	0	0

Logout

Prof Recommendation

Professor recommendation uses a machine learning model to suggest suitable instructors for a course based on factors like expertise, teaching history, student feedback, and availability. This helps administrators assign the most appropriate faculty to each course, improving teaching quality and student outcomes through intelligent, data-driven recommendations.



Navigation

Go to

○ Dashboard
○ Module
○ Assignment
○ Assesment
○ Course Recom
○ Prof Recom
○ Payment
● Feedback
○ Customer Care

Specialization: ds

course: ml

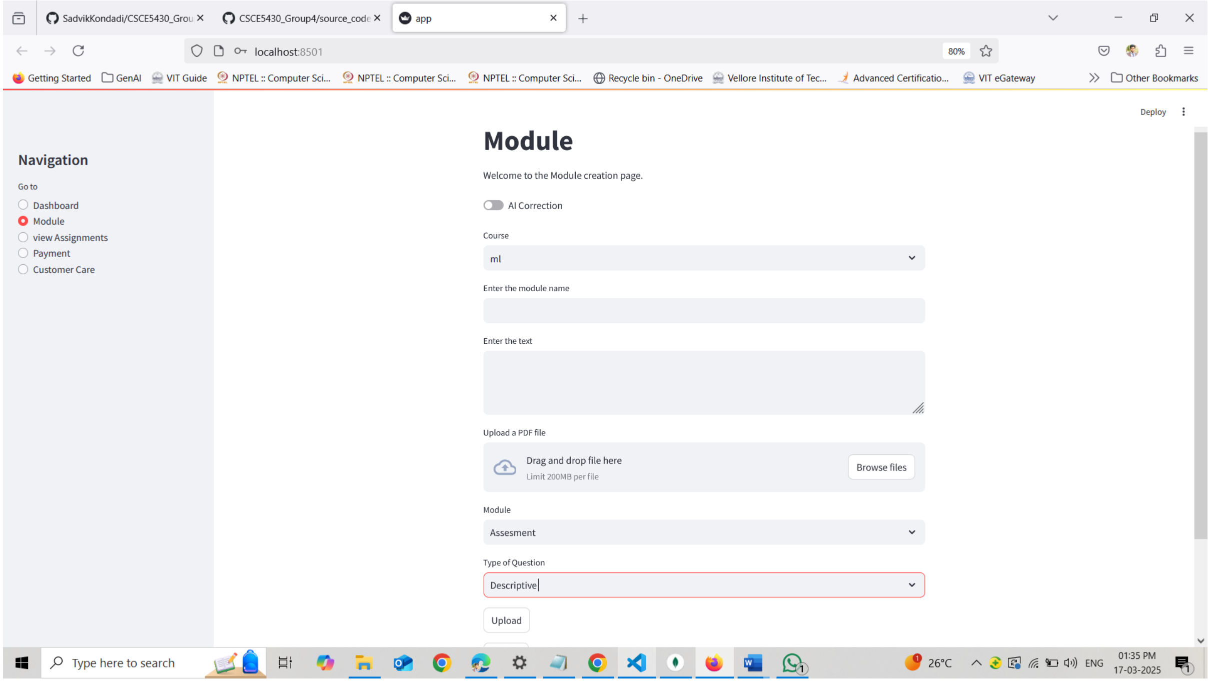
instructor: kris

Already Submitted

Logout

Feedback

The Feedback page allows users to share their opinions, suggestions, or concerns about courses, instructors, or the platform. It typically includes rating options and comment boxes, helping administrators and instructors improve the learning experience based on user input. This feature encourages transparency, engagement, and continuous improvement within the system.



Module

Welcome to the Module creation page.

AI Correction

Course: ml

Enter the module name

Enter the text

Upload a PDF file

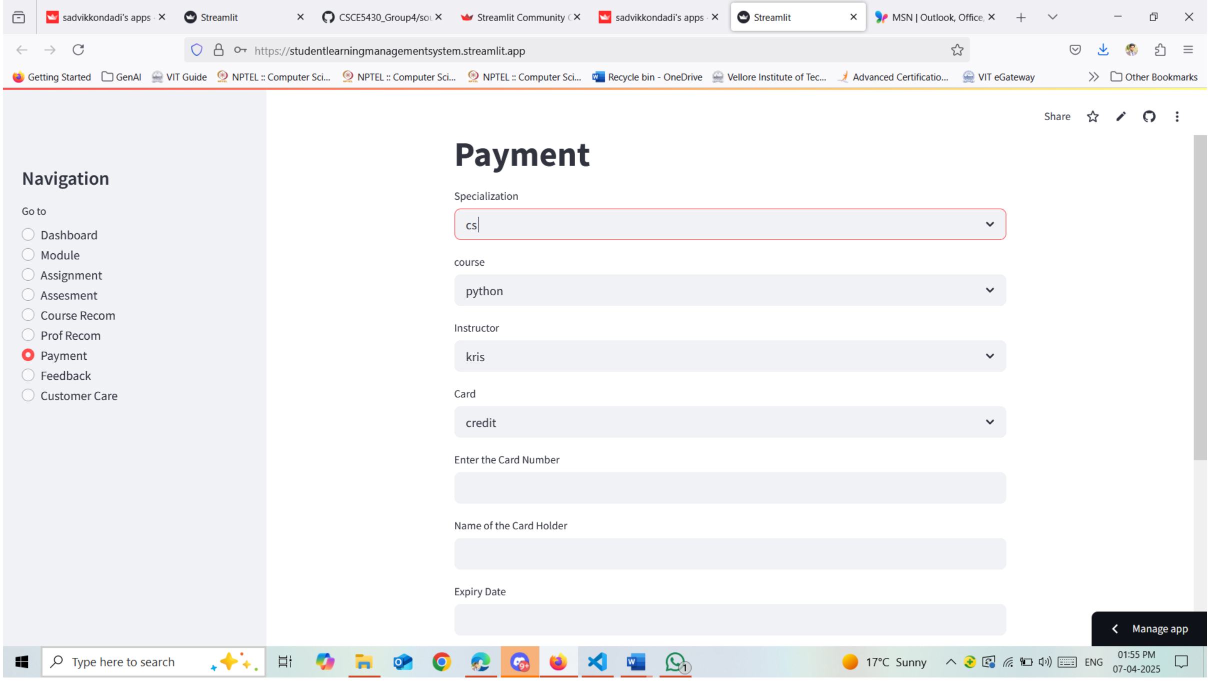
Drag and drop file here
Limit 200MB per file

Module: Assesment

Type of Question: Descriptive

Assessment Instructor View

The materials, like learning content, assignments, assessments, are uploaded by the Instructor in the module tab and can be downloaded and displayed to the student in the Module (Learning Content), Assignment, and Assessment tabs respectively. The Assessment questions can be of four types. They are Descriptive Questions, MCQ Questions, More than one Answer MCQ Questions, and True or False Questions.



Payment

Specialization: cs

course: python

Instructor: kris

Card: credit

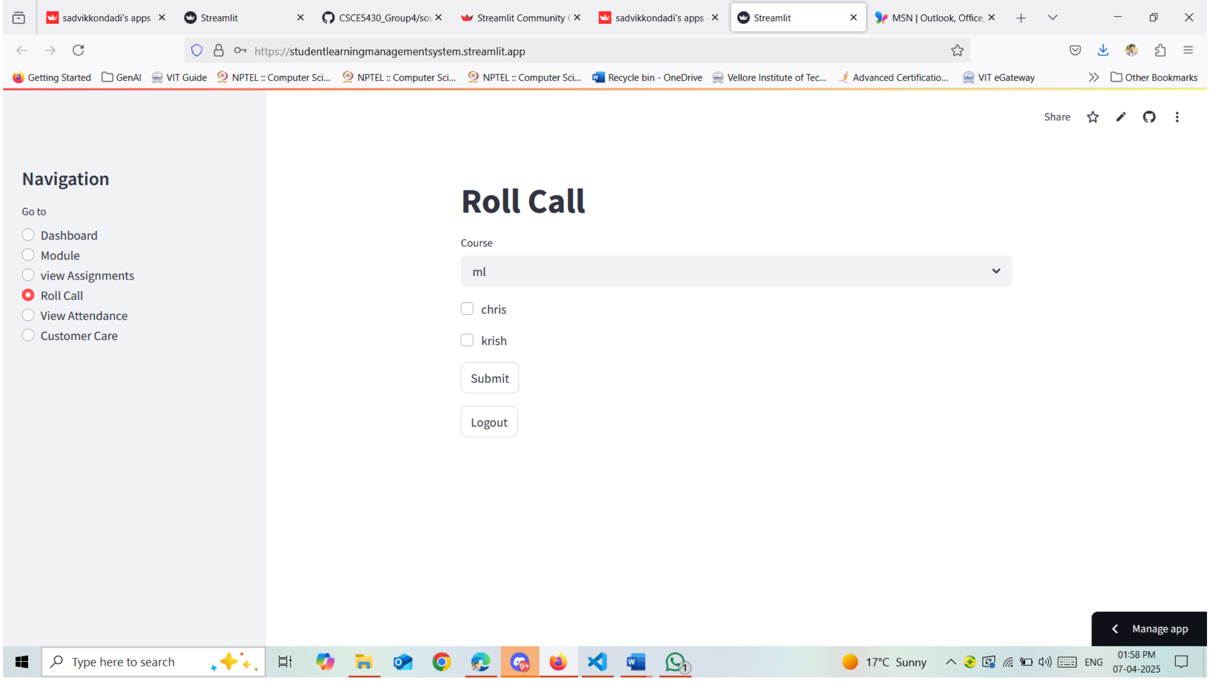
Enter the Card Number

Name of the Card Holder

Expiry Date

Payment

A payment system facilitates the transfer of money between buyers and sellers. It includes methods like credit/debit cards, mobile payments, and digital wallets. The system ensures secure transactions, manages authorization and settlement, and integrates with banks or payment gateways, providing a seamless way to complete financial exchanges.



Navigation

Go to

- ○ Dashboard
- ○ Module
- ○ view Assignments
- ● Roll Call
- ○ View Attendance
- ○ Customer Care

Roll Call

Course

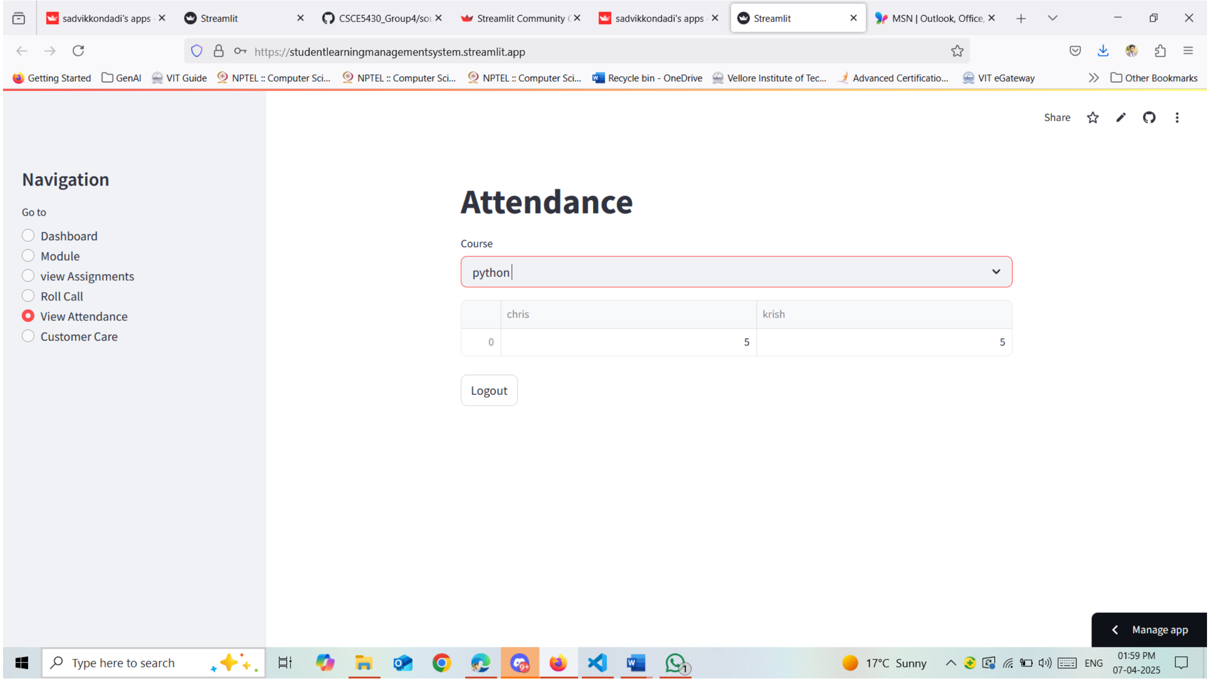
ml

□ chris

□ krish

Roll Call

A **Roll Call System** is a digital or manual method used to track attendance. It records the presence or absence of individuals (e.g., students, employees) in real-time. Often used in classrooms or meetings, it can be automated using software to streamline data collection, reporting, and reminders.



Navigation

Go to

○ Dashboard
○ Module
○ view Assignments
○ Roll Call
● View Attendance
○ Customer Care

Attendance

Course

python

	chris	krish
0	5	5

Logout

View Attendance

A **View Attendance System** allows users (e.g., students, teachers, or admins) to check attendance records. It displays attendance status (present, absent, or late) for each individual over a specified period. Features may include filtering by date, student, or class, and generating reports or summaries for analysis and tracking.

5.Instructions to Compile and Run the Code

- Download app.py code from Github, open in VS code and open terminal.
- The dependencies are installed by using the command ‘pip install -r requirements’ in the bash terminal.
- Run the program app.py by using the command 'streamlit run app.py' in the bash terminal.
- The default browser will open the Web UI.

6.Peer Review:

- In the last review session we discussed about phase-1 of our project and they given 2 feedbacks.
- One is to update the readme file as it not that much effective. So we chamged the readme file which explains everything about the project briefly.
- Second one is to keep each functionality files separately in git hub source code. But we thought as of now it will be easy to download execute the code easier. So in the last deliverable we will separate each files according to their functionality.
- As when it comes to functionality and execution they told everything is good and they too downloaded and executed the code and everything is fine.

7. Reflection:

- The Implementation of Customer Care almost took 5 hours but in the end, it worked successfully.
- The Payment can be improved by using of bank third party to pay by credit Cards.
- The Document Upload has bugs but at last, the PDF view is Implemented.

8. Member Contribution Table:

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Sadvik Kondadi	Professor recommendation System	11 %	
Sai Chandra Teja Akkala	Professor recommendation System	11%	
Shashank Kodishala	Assessment Enhancements	11%	
Deepak Reddy Punuru	Payment and Subscription management	11%	
Deepu gondhi	Content Management	11%	
Thanwish Ram Pothugunta	Attendance Tracking	11%	
Sanjay Ramaswamy Adla	Recommended Cources for Specilalization	11%	
Padmaja Soma	Feedback & Survey System	11%	
Nithin Reddy Pinikesi	Attendance Tracking	11%	